Author – *A.Kishore/Sachin*
http://appsdba.info

Output Post Processor (OPP) – An Overview

- Concurrent processing uses the Output Post Processor (OPP) to enforce post-processing actions for concurrent requests.

- Post-processing actions are actions taken on concurrent request output. An example of a post-processing action is that used in publishing concurrent requests with XML Publisher.

- The Output Post Processor (OPP) is an enhancement to Concurrent Processing and is designed to support XML Publisher as post-processing action for concurrent requests. If a request is submitted with an XML Publisher template specified as a layout for the concurrent request output, then after the concurrent manager finishes running the concurrent program, it will contact the OPP to apply the XML Publisher template and create the final output.

- A concurrent manager contacts an available OPP process when a running concurrent request needs an OPP processing action. Concurrent manager uses a local OPP process (that, is, on the same node) by default, but will choose a remote OPP if no local OPP process is available.

- There should always be at least one OPP process active in the system. If no OPP service is available, completed requests that require OPP processing will complete with a status of Warning.

- The Output Post Processor makes use of the Oracle Streams Advanced Queuing (AQ) database feature. Every OPP service instance monitors the FND_CP_GSM_OPP_AQ queue for new messages and this queue has been created with no value specified for primary_instance (link). This implies that the queue monitor scheduling and propagation is done in any available instance. In other words, ANY OPP service instance may pick up an incomming message independent of the node on which the concurrent request ran.

- Maximum Memory Usage Per Process:
  The maximum amount of memory or maximum Java heap size a single OPP process can use is by default set to 512MB. This value is seeded by the Loader Data File: $FND_TOP/patch/115/import/US/afoppsrv.ldt which specifies that the DEVELOPER_PARAMETERS is "J:oracle.apps.fnd.cp.gsf.GSMServiceController:-mx512m".

Determine the current maximum Java heap size:

```
SELECT service_id, service_handle, developer_parameters
FROM fnd_cp_services
WHERE service_id = (SELECT manager_type
FROM fnd_concurrent_queues
WHERE concurrent_queue_name = 'FNDCPOPP');
```

```
SERVICE_ID SERVICE_HANDLE DEVELOPER_PARAMETERS
---------- -------------- -----------------------------------------------------
1091 FNDOPP J:oracle.apps.fnd.cp.gsf.GSMServiceController:-mx512m
```

- Increase the maximum Java heap size for the OPP to 1024MB (1GB):

```
UPDATE fnd_cp_services
SET developer_parameters =
'J:oracle.apps.fnd.cp.gsf.GSMServiceController:-mx1024m'
WHERE service_id = (SELECT manager_type
FROM fnd_concurrent_queues
WHERE concurrent_queue_name = 'FNDCPOPP');
```

The OPP queue can be Recreated the using $FND_TOP/patch/115/sql/afopp002.sql file as 'APPLSYS' user. On running the script you will be prompted for username and password.

- There are two new profiles options that can be used to control the timeouts:

Profile Option : Concurrent:OPP Response Timeout
Internal Name : CONC_PP_RESPONSE_TIMEOUT
Description : Specifies the amount of time a manager waits for OPP to respond to its request for post processing.

Profile Option : Concurrent:OPP Process Timeout
Internal Name : CONC_PP_PROCESS_TIMEOUT
Description : Specifies the amount of time the manager waits for the OPP to actually process the request.